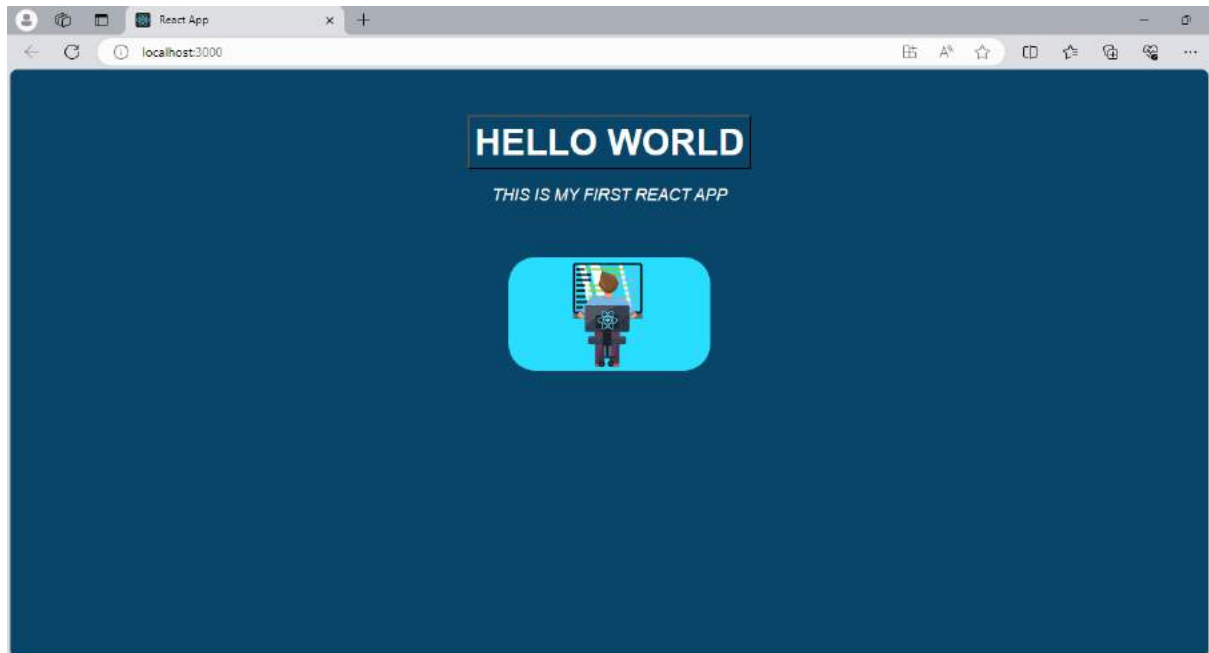

ASSIGNMENT 4

- Create a Hello World Web Page using React JS



App.js

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <button onClick={()=>{
        alert("WELCOME TO THE HOMEPAGE")
      }}> HELLO WORLD</button>
      <br></br>
      <p>THIS IS MY FIRST REACT APP</p>
      <div></img></div>
    </div>
  );
}

export default App;
```

App.css

```

.App {
  text-align: center;
}

.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

.App-link {
  color: #61dafb;
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}

```

Index.css

```

body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',
'Oxygen',
  'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
  sans-serif;

```

```
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}

*{
  font-size: larger;
  font-weight: bold;
  background-color: rgb(9, 69, 104);
  font-family: Arial, Helvetica, sans-serif;
  height: 60px;
  align-items: center;
  text-align: center;
  color: white;
}

.App{
  padding: 50px 50px;
}

p{
  font-weight: 300;
  font-size: large;
  font-style: italic;
}

img{
  border-radius: 30px;
  height: auto;
}
```

- Create a Sign-Up Form in React JS. The Sign-up form should ask for Name, Mobile, Email, Address, Date of Birth, Gender, Username, Password and Confirm Password. The Form should have two buttons, one for reset and one for submitting. The Submit Button should be enabled only when all the input values are validated.

App.js

```
//import logo from './logo.svg';
import './App.css';
import { useState ,useEffect} from 'react';
const initialFormData = {
  name: '',
  mobile: '',
  email: '',
  address: '',
  birthdate: '',
  gender: '',
  username: '',
  password: '',
  confirmPassword: ''
}

function App() {
  const [formData, setFormData] = useState(initialFormData);
  const [isValid, setIsValid] = useState(false);

  const handleSubmit = (e) => {
    e.preventDefault();
    // Submit form data
    console.log(formData);
    // Reset form
    setFormData(initialFormData);
    setIsValid(false);
  };

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData({
      ...formData,
      [name]: value
    });
  };

  const validateForm = () => {
```

```

    const { name, mobile, email, address, dob, gender, username, password,
confirmPassword } = formData;
    const isValidName = name?.trim() !== '';
    const isValidMobile = /^\d{10}$/.test(mobile);
    const isValidEmail = /^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email);
    const isValidAddress = address?.trim() !== '';
    const isValidDOB = dob?.trim() !== '';
    const isValidGender = gender !== '';
    const isValidUsername = username?.trim() !== '';
    const isValidPassword = password?.length >= 8;
    const isValidConfirmPassword = confirmPassword === password;

    setIsValid(
      isValidName &&
      isValidMobile &&
      isValidEmail &&
      isValidAddress &&
      isValidDOB &&
      isValidGender &&
      isValidUsername &&
      isValidPassword &&
      isValidConfirmPassword
    );
  };

  // Check form validity whenever formData changes
  useEffect(() => {
    validateForm();
  }, [formData]);

  const handleReset = () => {
    setFormData(initialFormData);
    setIsValid(false);
  };

  return (
    <div>
      <h2>Sign Up</h2>
      <form onSubmit={handleSubmit}>
        <div>
          <label htmlFor="name">Name:</label>
          <input type="text" id="name" name="name" value={formData.name}
onChange={handleChange} placeholder="Enter your name" />
        </div>
        <div>
          <label htmlFor="mobile">Mobile:</label>

```

```
        <input type="text" id="mobile" name="mobile" value={formData.mobile}
onChange={handleChange} placeholder="Enter your mobile number" />
    </div>
    <div>
        <label htmlFor="email">Email:</label>
        <input type="email" id="email" name="email" value={formData.email}
onChange={handleChange} placeholder="Enter your email address" />
    </div>
    <div>
        <label htmlFor="address">Address:</label>
        <input type="text" id="address" name="address"
value={formData.address} onChange={handleChange} placeholder="Enter your
address" />
    </div>
    <div>
        <label htmlFor="dob">Date of Birth:</label>
        <input type="date" id="dob" name="dob" value={formData.dob}
onChange={handleChange} />
    </div>
    <div>
        <label htmlFor="gender">Gender:</label>
        <select id="gender" name="gender" value={formData.gender}
onChange={handleChange}>
            <option value="">Select</option>
            <option value="male">Male</option>
            <option value="female">Female</option>
            <option value="other">Other</option>
        </select>
    </div>
    <div>
        <label htmlFor="username">Username:</label>
        <input type="text" id="username" name="username"
value={formData.username} onChange={handleChange} placeholder="Enter your
username" />
    </div>
    <div>
        <label htmlFor="password">Password:</label>
        <input type="password" id="password" name="password"
value={formData.password} onChange={handleChange} placeholder="Enter your
password" />
    </div>
    <div>
        <label htmlFor="confirmPassword">Confirm Password:</label>
        <input type="password" id="confirmPassword" name="confirmPassword"
value={formData.confirmPassword} onChange={handleChange} placeholder="Confirm
your password" />
    </div>
</div>
```

```

        <button type="submit" disabled={!isValid}>Submit</button>
        <button type="button" onClick={handleReset}>Reset</button>
      </div>
    </form>
  </div>
);
};
export default App;

```

index.css

```

body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',
'Oxygen',
  'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
  sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
  monospace;
}

input,textarea{
  color: black;
  background-color: #e0e0e0;
  width: 100%;
  border: 1px solid rgb(240, 237, 237);
  height: auto;
  padding: 10px 10px;
}

button{
  cursor: pointer;
  background-color: rgb(31, 145, 238);
  padding: 10px 15px;
  border: 5px solid whitesmoke;
}

button:hover{
  background-color: rgb(6, 211, 91);
  border: 2px solid white;
}

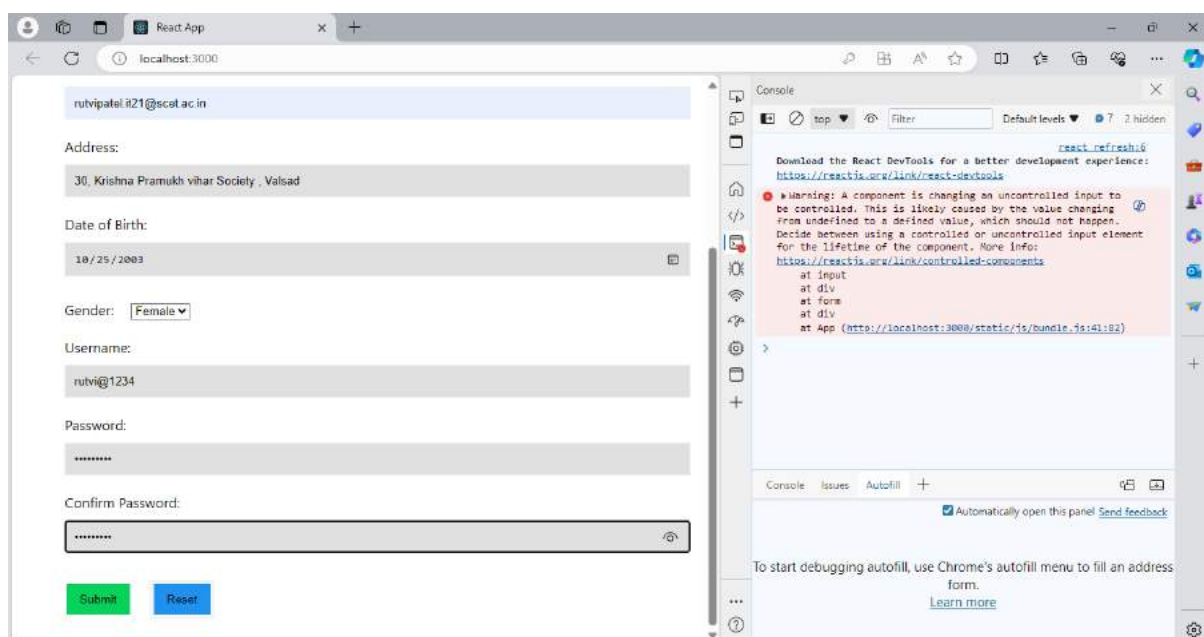
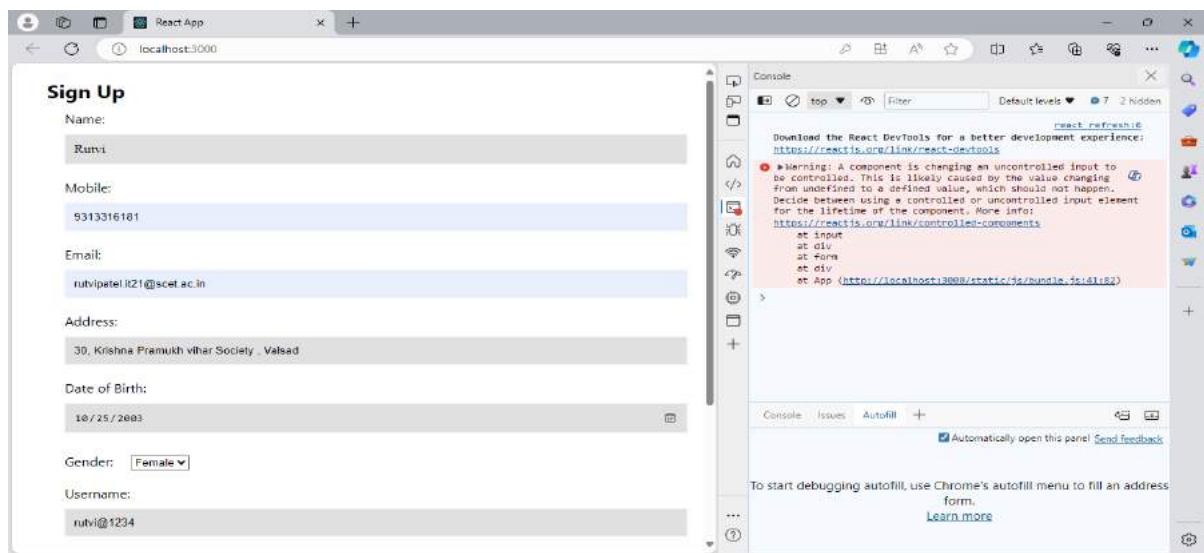
*{

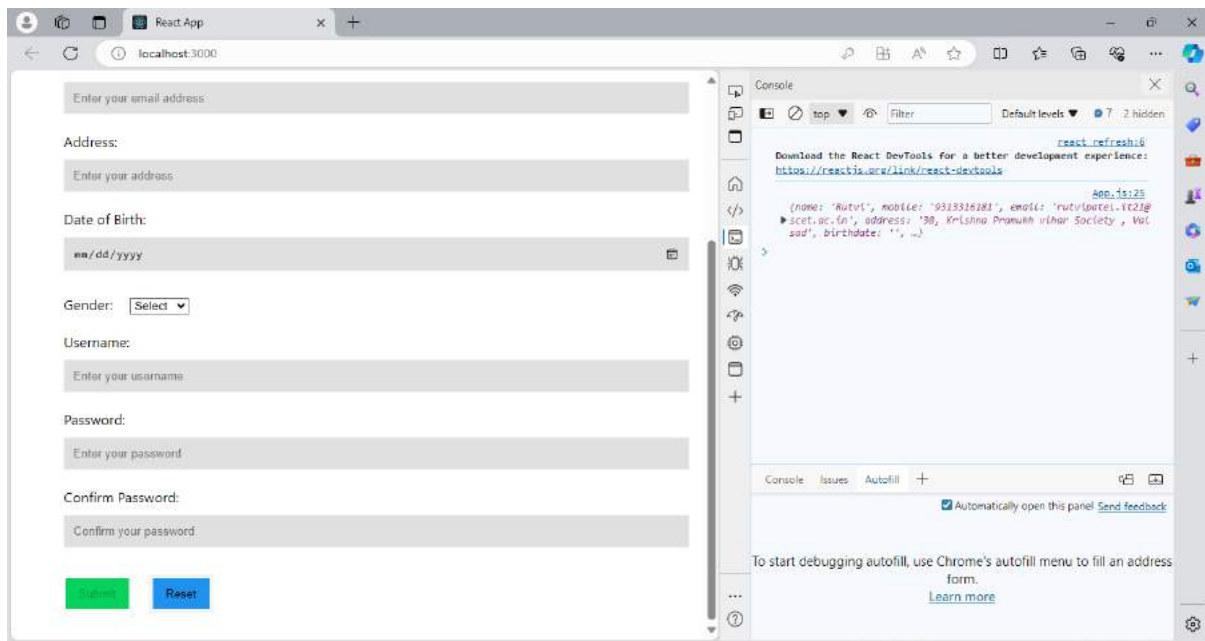
```

```

text-align: left;
background-color: white;
margin: 10px;
}
textarea{
  color: rgb(148, 148, 144);
  background-color: #e0e0e0;
}
.gender{
  font-size: large;
padding: 0%;
align-items: start;
}

```





- Develop a simple application using React.js where a user can Add/Delete notes. Each note timestamped as well.

App.js

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <button onClick={()=>{
        alert("WELCOME TO THE HOMEPAGE")
      }}> HELLO WORLD</button>
      <br></br>
      <p>THIS IS MY FIRST REACT APP</p>
      <div></img></div>
    </div>
  );
}

export default App;
```

index.css

```
.App {
  font-family: Arial, sans-serif;
  margin: 0 auto;
  padding: 20px;
  max-width: 600px;
}

.input-container {
  margin-bottom: 20px;
}

input[type="text"] {
  width: 70%;
  padding: 10px;
  font-size: 16px;
}

button {
  padding: 10px 20px;
  font-size: 16px;
  cursor: pointer;
  background-color: #007bff;
  color: white;
  border: none;
  border-radius: 5px;
}

.notes-container {
  display: grid;
  grid-gap: 10px;
}

.note {
  border: 1px solid #ccc;
  padding: 10px;
  border-radius: 5px;
}

.timestamp {
  font-size: 12px;
  color: #888;
}
```

